AR-006-756

ELECTRONICS RESEARCH LABORATORY

Communications Division

RESEARCH REPORT
ERL-0568-RR

CONSTANT MODULUS ALGORITHM TECHNIQUES FOR THE ENHANCEMENT OF SIGNAL QUALITY

by

Peter L. Trudinger

SUMMARY

The Constant Modulus Algorithm is an adaptive algorithm for selecting equalizer filter weights which attempts to correct for degradation in signal quality suffered by constant envelope signals (eg. FM) over an imperfect transmission channel. This report reviews and evaluates the standard CMA and several of its variants, which were designed to improve on its performance or extend its applicability. Two variants seem to significantly increase convergence rate and appear worthy of further study or implementation.

AUG 91 COPY No.

APPROVED FOR PUBLIC RELEASE

POSTAL ADDRESS: Director, Electronics Research Laboratory,PO Box 1600, Salisbury, South Australia, 5108.

CONTENTS

FIGURES

1 INTRODUCTION

A signal transmitted over a channel may be degraded in quality through the action of three factors - background noise, uncorrelated co-channel interference, and propagation phenomena, of which the most significant is multipath. If the quality of the signal is to be improved after reception, then an attempt must be made to filter out the effects of these factors as much as possible. Usually the design of such filters takes into account some knowledge (or assumptions) about the nature of the signal, channel, or interferers. The Constant Modulus Algorithm (CMA) is an attempt to compensate for interference and multipath effects on signals which have a constant envelope, such as frequency modulated signals. It is well known that the effect of additive interference or a multipath propagation channel on signals of constant modulus is to destroy the constant modulus and induce variations in modulus and phase. The idea behind the CMA is relatively simple: Since the original signal had a constant envelope, filter the received signal in such a way that the filter output also has a constant modulus. This process may produce a cleaner version of the signal of interest (SOI).

The Constant Modulus Algorithm was first proposed by Treichler and Agee in [1], and subsequently has been further investigated by Treichler, Larimore, and others. At about the same time as the CMA was developed, a similar approach was applied by Godard to the problem of blind equalization of modems [2]. Since then there have been several works on the suitability of the CMA as a blind equalizer. Others have also attempted to modify the original CMA to improve performance, or extend it to other classes of signals. This paper is a review of many of the published articles on the CMA.

The basic assumption in the CMA is that the the SOI has a constant modulus. In the case of multipath interference, it is also assumed that the multipath propagation channel can be modeled as a finite duration impulse filter, where the effective impulse response duration is greater than the reciprocal of the bandwidth of the SOI ("frequency selective multipath"). No assumptions are made in general about the nature of the noise of uncorrelated interferers, although, as will be mentioned later, these factors can drastically affect the success of the algorithm, resulting in the filter nulling the SOI, and selecting an interfering signal, or noise.

The CMA is based on two theoretical tools:

1. A measure of how much a signal varies from having a constant modulus
2. An optimization procedure, which minimizes this metric.

The filter is made adaptive by repeated applications of the optimization procedure. In the next two sections we shall examine in order these tools as they were originally proposed by Treichler and Agee. Section 4 will discuss extensions of the original CMA to other situations, and Section 5 will consider variations on the original optimization procedure. Section 6 is a conclusion.

2 THE CMA COST FUNCTION

The CMA was first formulated for an adaptive digital transversal FIR filter acting on the quadrature sampled distorted signal, as in Figure 1.

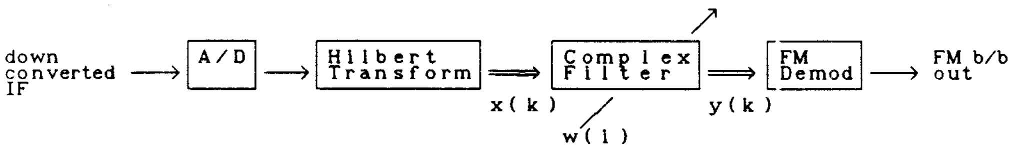


Figure 1. Block diagram of CMA (from [1])

We shall use the following notation:

N is the filter length

$\mathbf{x}(k)$ is the data vector of the N most recent signal samples

$$\mathbf{x}(k) = [x(k)\ x(k-1)\ \ldots\ x(k-N)]$$

$\mathbf{w}(i)$ is the complex weight vector for the filter at step i

y(k) is the filter output at time k

$$y(k) = \mathbf{x}(k)^T\mathbf{w}(i)$$

The values of i and k need not be the same, for example, the sampling rate might outstrip the processing speed for the weight update, or the algorithm might use a block of data for each update. However, for simplicity, we assume that the weights will be updated with each new sample, so i=k.

The aim of the CMA is to choose the weights $\mathbf{w}(k)$ so as to minimize the difference between the modulus of the output $|y(k)|$ and a constant, δ say. This requires defining a metric to measure the distance between two signals. The usual choice is a function of the form

$$J_{pq} = < (|y(k)|^p - \delta^p)^q >$$

where < . > denotes the expectation [3]. For computational simplicity, p and q are chosen to be small integers, either 1 or 2. The value of δ is arbitrary, and represents the gain of the output signal. It is usually set at 1.

3 THE OPTIMIZATION PROCEDURE

In the original CMA, the minimization of J was carried out using the method of steepest descent (SD-CMA). The weight update equation is

$$w(k+1) = w(k) - \mu \nabla_w J_{pq}(k)$$

where μ is the step size, ∇_w is the gradient operator with respect to w and $J_{pq}(k)$ is the cost function at the kth step. It was shown in [3] that

$$\nabla_w J_{pq}(k) = < q\ p\ \mathbf{x}^*(k)\ y(k)\ |y(k)|^{p-2}\ (|y(k)|^p - \delta^p)^{q-1} \\ (\mathrm{sgn}(|y(k)|^p - \delta^p))^q >$$

provided $|y(k)| \neq 0$. In practice, the expected value must be estimated. This is conveniently done using the point values at the kth step. The weight update equation then takes the form

$$w(k+1) = w(k) - \mu\ \mathbf{x}^*(k)\ \varepsilon_{pq}(k)$$

for appropriate functions ε_{pq}. For the four choices of p and q the weight update equations are:

(1,1) $$w(k+1) = w(k) - \mu\ \mathbf{x}^*(k)\ \frac{y(k)}{|y(k)|}\ \mathrm{sgn}(|y(k)| - \delta)$$

(2,1) $$w(k+1) = w(k) - 2\ \mu\ \mathbf{x}^*(k)\ y(k)\ \mathrm{sgn}(|y(k)|^2 - \delta^2)$$

(1,2) $$w(k+1) = w(k) - 2\ \mu\ \mathbf{x}^*(k)\ \frac{y(k)}{|y(k)|}\ (|y(k)| - \delta)$$

(2,2) $$w(k+1) = w(k) - 4\ \mu\ \mathbf{x}^*(k)\ y(k)\ (|y(k)|^2 - \delta^2)$$

Of these, the last (p = 2, q = 2) is the easiest to calculate with, since it does not involve a square root or discontinuous signum function. In their original paper [1] Treichler and Agee chose as the CMA cost function

$$J = \frac{1}{4} < (|y(k)|^2 - 1)^2 >$$

In this case, the weight update equation becomes

$$\begin{aligned} w(k+1) &= w(k) - \mu\ \mathbf{x}^*(k)\ y(k)\ (|y(k)|^2 - 1) \\ &= w(k) - \mu\ \varepsilon(k)\ \mathbf{x}^*(k) \end{aligned}$$

where

$$\varepsilon(k) = y(k)\ (|y(k)|^2 - 1)$$

The disadvantage of this choice of p and q is that the cost function and update equation involve higher powers of the data and so would require a larger word length for implementation. It would appear that different choices of cost function are suited to different situations.

With any iterative algorithm, there is the question of whether convergence will occur at all. It has been proven that, under certain assumptions, the most significant of which is that the filter length be sufficiently long, the standard complex SD-CMA with $p = q = 2$ will converge to a stationary point on the error surface, where $J' = 0$ (or to a point on the constraint boundary, if there is one) [4].

In their original paper Treichler and Agee reported on simulations of the CMA in various situations. In one set of simulations, they generated a signal by using band limited noise to modulate a 0 Hz complex carrier. Simple, one bounce, multipath was added using a linear shift invariant filter. A notch was created in the spectrum of the signal, and in some cases, circularly Gaussian white noise was added. The resulting signal was processed using a 256 weight filter and a small μ. Some of their results are reproduced in Figures 2 and 3.

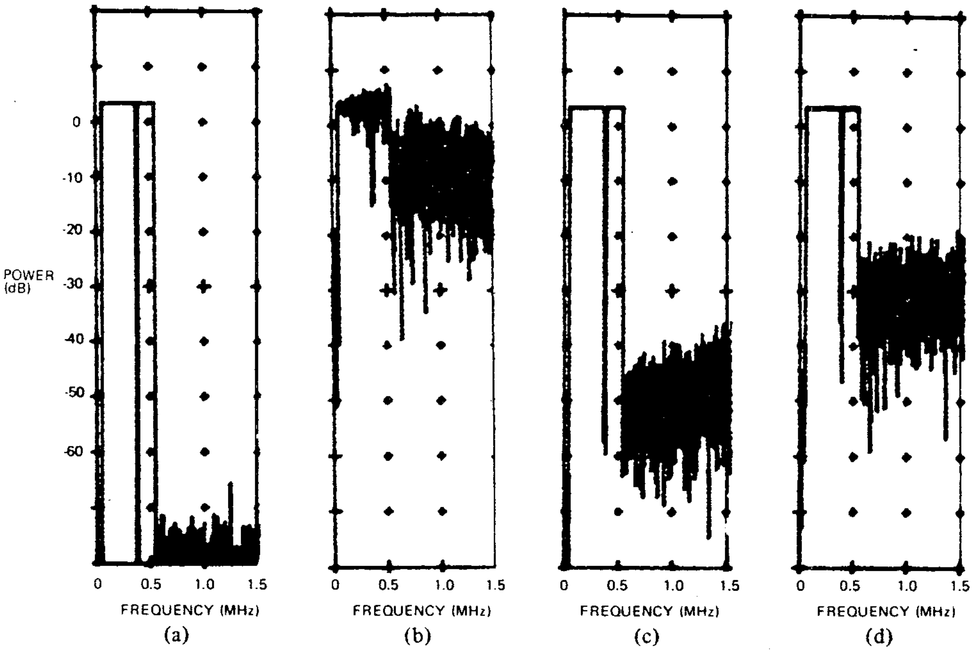


Figure 2. A comparison of baseband spectra at various points in the processing chain. (a) Generated baseband; (b) demodulated w/o multipath correction; no additive noise; (c) demodulated w/multipath correction; no additive noise; (d) demodulated w/multipath correction; additive noise present. (From [1])

The ability of the algorithm to recover the initial signal, as evidenced by Figure 2, is impressive. However, Figure 3 shows that many thousands of iterations are needed to achieve this. Whilst some of the slowness of convergence can be explained by the choice of a small adaptation constant, other published results suggest that the convergence rate of the CMA is inherently slow, and thousands of iterations may be required, even in simple cases. Indeed in [5] it is claimed that, under realistic multipath conditions, 200,000 iteration steps and more have to be carried out to obtain an approximation of the optimum weight vector, and further, that 500 to 1000 coefficients may be needed. In [3] and [6] Larimore and Treichler derived formulae to predict the asymptotic convergence rate of the CMA for sinusoidal inputs. They derived an approximate time constant for the CMA and showed that it varied inversely with the step size and inversely with the square of the signal amplitude.

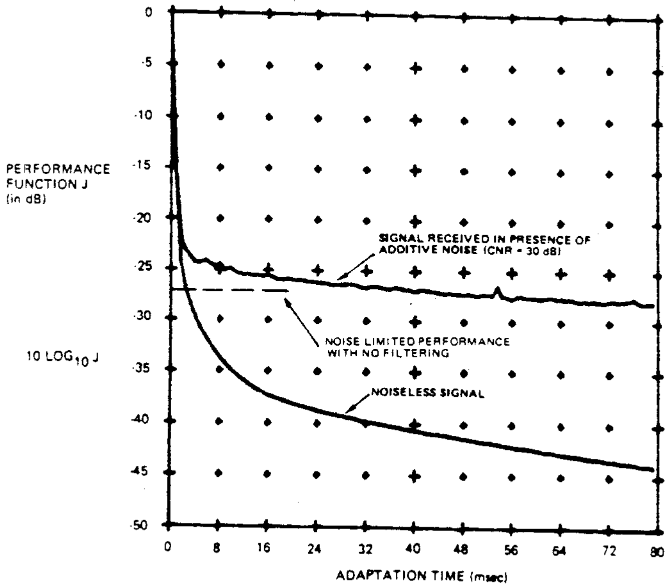


Figure 3. "Learning curve" of the CMA for a specific signal and multipath condition: 10^4 iterations per ms (from [1]).

Larimore and Treichler have also reported a condition they call "noise capture" for a signal plus noise [7]. If the noise level is too high (they use, for example, -4dB) the algorithm may not converge to the signal in any reasonable period of time. Intuitively, they explain this in terms of the filter response to the strongest signals. The filter evolves fastest for frequencies where the input energy is strongest. So if the filter is initialized with the modulus of the output too large, it tends to reduce its gain fastest at the frequencies of the SOI. They claim that mathematical analysis suggests that the cost function surface has a large flat area, which traps the algorithm for a long period. Agee, however, has disagreed with this explanation. In [8], he performed a careful analysis of the cost function of the CMA and of more general modulus restoring algorithms for the case of an array, and concluded that, in general, one could expect two types of stationary behaviour - either convergence to the SOI (signal capture) or convergence to a state which nulls the SOI (noise capture).

The poor convergence rate of the CMA has prompted several researchers to modify the algorithm to improve convergence. We shall discuss these modifications later.

Global convergence, however slow, is not enough. It is also necessary for the algorithm to output the SOI. Unfortunately, the standard SD-CMA is not well behaved in this respect. Even in the simplest cases, the cost function has a non-convex surface. On a trivial level, $w(k) \equiv 0$ and $y(k) \equiv 0$ is an equilibrium point, albeit unstable. (Incidentally, this means that the initial weight vector must be chosen to be non-zero. One obvious choice is to set all but one of the weights to be zero.) There are also other sources of non-uniqueness for the SD-CMA. Since the cost function ignores any phase information, the output of the algorithm can, at best, only be unique up to a constant phase shift. This can be corrected for using a phase locked loop. It is also possible that two optimal solutions could be equal, except for a constant group delay. This situation could come about if the filter length was longer than necessary, with 0s at the head or tail, so that an optimal weight sequence could be shifted to different locations on the delay line.

These cases of non-uniqueness are not serious on a practical level, and would not affect the performance of the algorithm in a multipath situation.

However, in an environment where two or more uncorrelated signals are present a more serious form of non-uniqueness may occur. The algorithm may be captured by an interferer, that is, application of the SD-CMA may result in one of the unwanted signals being output by the filter. Furthermore, it seems difficult to predict how the CMA will behave in such an environment. In [9] Larimore and Treichler examined the capture phenomenon in what is possibly the simplest case, namely, of two sinusoids and no noise. They derived a coupled system of equations for the gains of each signal at each time step, and from these were able to describe regions where the result of the algorithm could be predicted. Figure 4 shows the approximate boundaries between the regions where the algorithm locks onto the SOI or is captured by an interferer, for different values of the initial output powers of the two signals (expressed as the ratio of the powers of the signal to the interference, i.e., the SIR). Figure 5 shows some of the trajectories followed by the gains as the algorithm evolves, for two different SIRs.

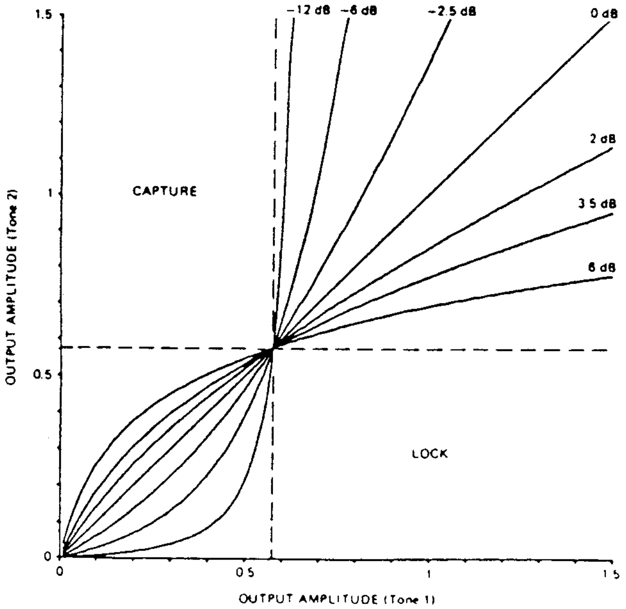
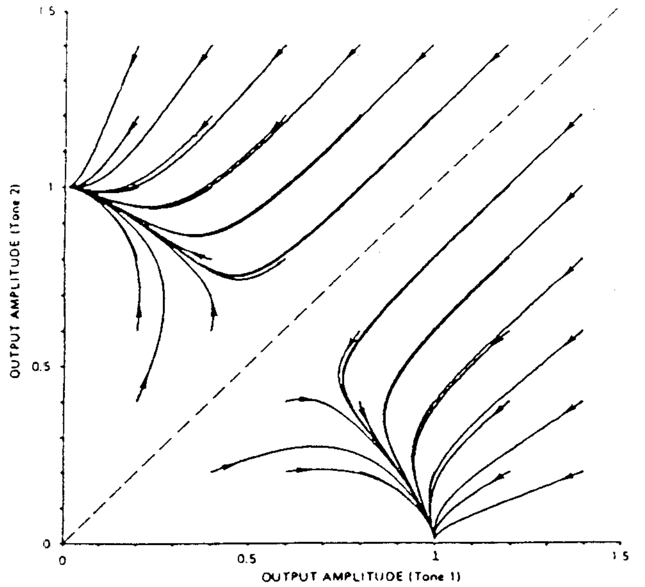
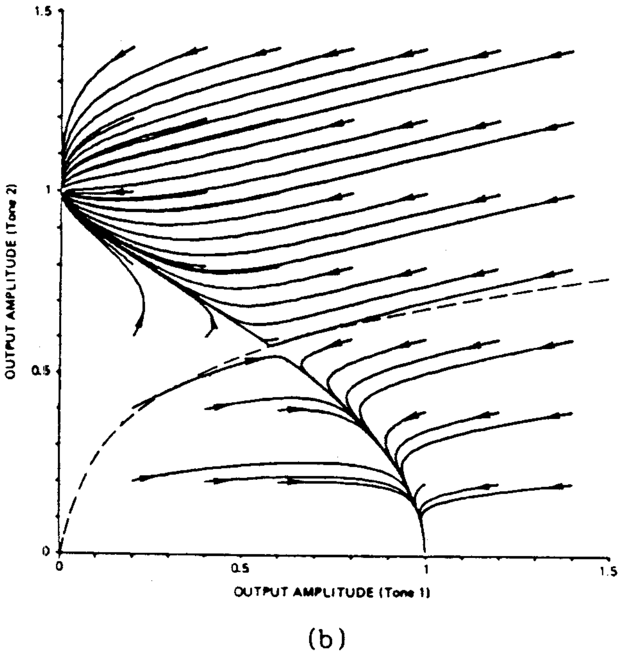


Figure 4. Lock and capture zone boundaries as a function of signal to interference ratio (SIR). The SOI is sinusoid 1, the interferer sinusoid 2. (from [9])



(a)



(b)

Figure 5. Output amplitude behaviour for different SIR. (a) SIR = 0 dB; (b) SIR = 6 dB. The dashed line is the boundary between lock and capture, c.f. Fig. 4. (from [9]).

It should be noted that these figures display output amplitude. This is a function of the received power of the signals, and the filter state. Thus the initial state of the filter is crucial. It would appear that a simple rescaling of the initial choice of weights can alter the behaviour from lock to capture, regardless of the input power ratio of the signals. The situation would, of course, be far more complicated in a real environment.

The concept of using a single linear time-invariant filter has inherent shortcomings when used to null an uncorrelated interferer. The filter can only null the interferer by forcing the gain to low levels in the spectral bands of the interferer, and this could result in a loss of signal quality. This may occur regardless of the algorithm chosen. Different algorithms will result in different tradeoffs between the presence of signal and interferer at the output [10]. It would seem from simulations that in the presence of an interferer, the SD-CMA filter tends to a bandpass or notch design, whilst in a multipath environment, the filter approaches a design that tends to correct for the multipath induced phase and amplitude variations [9]. Carrying the first observation to an extreme, one would expect that the CMA would not be able to suppress a wideband interferer, without also suppressing the SOI.

The CMA bears a resemblance to the Least Mean Squares Algorithm [11]. For example, for p = 1, q = 2, the CMA is very like the LMS algorithm, with reference signal $y(k)/|y(k)|$ (see Figure 6). However, in the case of the CMA, the external reference signal is replaced by external knowledge about the properties of the transmitted signal. The similarities between the algorithms appears to extend to their properties, e.g., the convergent response of a CMA based filter appears to be very Wiener like [1, 10]. Because of this similarity, the LMS algorithm can be used as an intuitive aid to understanding the CMA. Also variants of the LMS algorithm can serve as the motivation for modifying the CMA to improve performance.

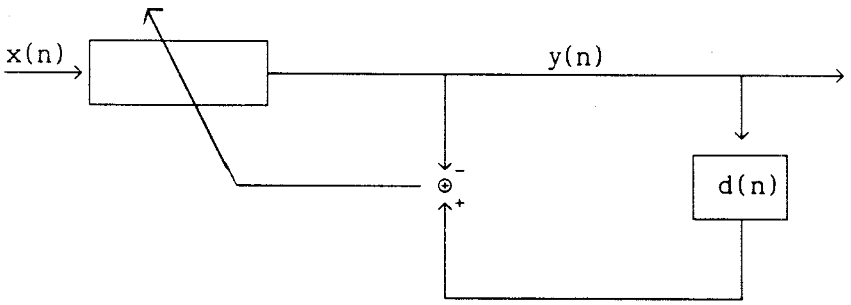


Figure 6. Block diagram illustrating similarity between LMS and CMA. In the LMS d(n) is the reference signal, in the CMA y/‖y‖ for p = 1, q = 2, etc. (from [11])

4 EXTENSIONS OF THE STANDARD CMA

In this section we will review some of the schemes proposed to extend the standard steepest descent CMA to situations other than the processing of a single complex constant modulus signal. Since the optimization method is not altered, a slow convergence rate would be expected of these algorithms. Capture effects could also appear.

4.1 The Real CMA

The SD-CMA is formulated for processing a complex signal. In practice, however, it is only the real signal which is of interest in almost all circumstances. It is most likely that the receiver would deliver a real signal, which would then be Hilbert transformed to produce the input for the CMA, and that only the real part of the output of the CMA would be used. Since processing complex signals increases the complexity and cost of hardware, there would seem to be some benefit in reducing the number of complex operations performed in the CMA. This issue was addressed by Larimore and Treichler in [12 and 13]. They split the various steps in the algorithm into real and imaginary parts, and discarded those quantities which did not affect the real part of the output. From this analysis, they concluded that the only necessary complex operation was in the calculation of ε(k), where the complex value of y(k) was needed. This complex signal could be provided by a Hilbert transformer, and a delay to compensate for the length of the Hilbert transform. A block diagram for this is given in Figure 7.

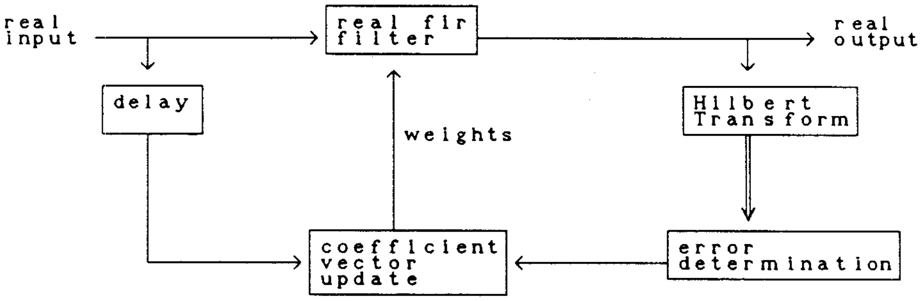


Figure 7. CMA using only real arithmetic (from 12)

The obvious question to ask next is: what would be a satisfactory length for the Hilbert transform? Surprisingly, Larimore and Treichler reported that their simulations showed that the Hilbert transform could be shortened to have length one (that is, omitted altogether) without any significant performance degradation. The filter tends to the same frequency response even if the Hilbert transform is imperfect or absent. The effect of shortening or omitting the transform is to give a final filter gain different from one, and noisier weight updates. The equations for the real CMA are:

$$y(k) = \mathbf{x}^T(k)\ \mathbf{w}_r(k)$$

$$\mathbf{w}_r(k+1) = \mathbf{w}_r(k) - \mu\ \varepsilon_r(k)\ \mathbf{x}(k)$$

$$\varepsilon_r(k) = (|y(k)|^2 - 1)\ y(k)$$

where **x** and y are real. A block diagram is given is Figure 8

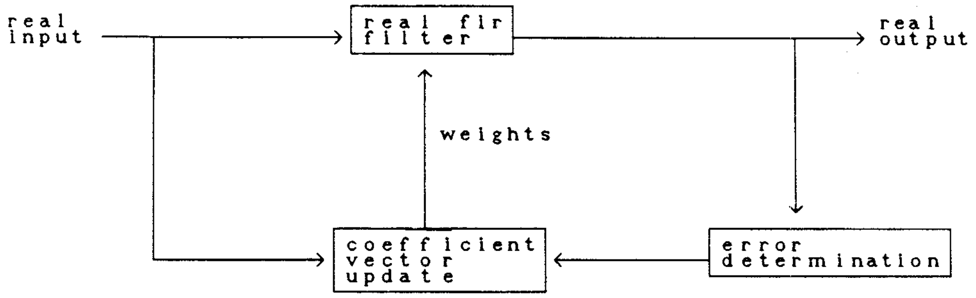


Figure 8. CMA employing real arithmetic and simplified error determination (from [12]).

Convergence of the real CMA was investigated theoretically in [4] and [14]. Friedlander and Smith in [4] claim that the real CMA as formulated above has asymptotic bias and propose a modified and unbiased version of the real CMA. The analysis in that paper requires that certain conditions be met at each instant of time. Johnson, Dasgupta and Sethares in [14] took a different approach, and analyzed the average behaviour of the real CMA. They proved local exponential stability of the averaged real CMA. Whilst allowing for the weight jitter noted by Larimore and Treichler, this effectively implies the convergence of the real CMA and its robustness, for example to noise or mismodelling.

4.2 Known Modulus Algorithm

Somewhat confusingly, there have been two extensions to the CMA for which the name Known Modulus Algorithm (KMA) has been suggested.

In [15] Ferrara proposed a version of the CMA for the situation where the absolute level of the envelope of the SOI is known or measurable. Essentially, if this modulus is d, then his KMA involves minimizing $< (|z(k)|^2 - d)^2 >$, where z is the complex output signal associated with $\mathbf{w}^T\mathbf{x}(k)$, **w** and **x** real, with the constraints that the centre weight of the filter is held at unity and the weights are symmetric about the centre. These constraints protect the filter from being captured by a constant modulus interferer with amplitude other than d. The algorithm was presented in the context of real signals, but did not eliminate the Hilbert transform. Ferrara also suggested that if no amplitude information was available, the value of d could be varied, until the algorithm locked on to a signal. If this was not a SOI, the process of "tuning" d could be resumed.

A more general treatment of the means by which constraints can be incorporated into the CMA will be discussed below.

Another Known Modulus Algorithm was put forward by Larimore and Treichler in [13]. In their scenario, the SOI had a varying, but known amplitude. Such a signal might be the result of amplitude shaping of each baud or symbol, in order to control the signal power spectrum. In this situation, the cost function is

$$J = \frac{1}{2} < (|y(k)|^2 - m^2(k)) >$$

and

$$\varepsilon(k) = (|y(k)|^2 - m^2(k))\ y(k)$$

where m(k) is the known modulus of the transmitted signal. Some sort of synchronization is required in general. However, if the baud interval is an integer multiple of the sampling interval, the algorithm is self synchronizing, as it will attempt to place the correct group delay on y(k) to match the template m(k). The special shape of the envelope of the transmitted signal reduces the likelihood of capture.

Modifications of the CMA along these lines have also been proposed which are applicable to the problem of blind equalization of multiple modulus signals [16]. If the transmitted source is highly correlated and non-uniform, the standard CMA does not converge to a useful solution. Two variations were proposed by Sethares, Rey and Johnson in [16]. One was to incorporate the various moduli in the point estimate of the cost function as

$$J = (y^2(k) - M_1)^2\ (y^2(k) - M_2)^2\ \ldots(y^2(k) - M_n)^2$$

The other was to form an hybrid of the CMA and Decision Directed algorithms, that is, to have n cost functions

$$J_i(k) = (y^2(k) - M_i)^2 \qquad , i = 1,2,\ldots,n.$$

and one update

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu\ y(k)\ \min_i[(y^2(k) - M_i)^2]\ \mathbf{x}(k)$$

where the "min" function compares the reconstructed signal to the nearest M at each time k. The authors preferred the latter algorithm.

4.3 Array CMA

The CMA can be extended in a straightforward fashion to give a beamforming algorithm for an array of receivers [13, 17]. In this situation, the weight and signal vectors for each antenna are concatenated. If there are n array elements, with vectors of sampled data $\mathbf{x}_i(k)$, $i = 1,2, \ldots, n$, and corresponding weight vectors $\mathbf{w}_i$, then we set

$$\mathbf{x}(k) = \mathbf{x}_1(k)\oplus\mathbf{x}_2(k)\oplus\ \ldots\ \oplus\mathbf{x}_n(k)$$

$$\mathbf{w}(k) = \mathbf{w}_1(k)\oplus\mathbf{w}_2(k)\oplus\ \ldots\ \oplus\mathbf{w}_n(k)$$

$$\begin{aligned} y(k) &= \mathbf{x}^T(k)\mathbf{w}(k) \\ &= \mathbf{x}_1^T(k)\mathbf{w}_1(k) + \mathbf{x}_2^T(k)\mathbf{w}_2(k) + \ldots + \mathbf{x}_n^T(k)\mathbf{w}_n(k) \end{aligned}$$

The cost function and weight update equation are the same as for the standard CMA. Larimore and Treichler also proposed a polarization combiner for a two element array using the CMA in [13], for separating two signals with the same frequency, but different polarizations. Although their simulations demonstrated the feasibility of this scheme, it has been claimed in [18] that the specific multistage structure put forward by them had maladjustment problems between the stages.

4.4 Linearly Constrained CMA

The Known Modulus Algorithm of Ferrara added some simple linear constraints to the CMA to force the algorithm to converge to solutions with a prescribed modulus. In other circumstances, linear constraints might be formulated to include some *a priori* information about the signal of interest, e.g., the weights in a single receiver system might be constrained to favour an FM signal over a CW interferer, by having unity gain at the FM carrier frequency, or the weights in an array could be constrained to compensate for unknown or changing array geometry. Rude and Griffiths in [19] have shown how a constant modulus problem with a set of linear constraints may be converted into an unconstrained CMA optimization problem, using a preprocessor called a generalized sidelobe canceler. Their method is based on the well known property that a solution of a nonhomogeneous system of linear equations can be expressed as a particular solution plus the solutions of the corresponding homogeneous system. They split the weight vector into two parts. One part, which is fixed, guarantees that the output vector y will satisfy the constraints. The optimization takes place over the other part.

At this point, one of the basic inconsistencies amongst practitioners of signal processing rears its head, namely, whether to use the real transpose or hermitian (conjugate) transpose. Treichler, Agee, and most other CMA researchers chose the former. Rude and Griffiths use the latter. They express the output of the filter as

$$y = \mathbf{w}^H \mathbf{x}$$

The main effect of this, apart from confusing the unwary reader, is that the weights of Rude and Griffiths are the complex conjugates of the weights for the others, e.g., if the cost functions for the unconstrained CMA is

$$\frac{1}{4} < (|y|^2 - \delta^2)^2 >$$

then the update equation for the Hermitian transpose version of the CMA is

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \, (|y|^2 - \delta^2) \, y^* \, \mathbf{x}$$

which is the conjugate of the update equation for the real transpose version. We shall express the LC-CMA in terms of the real transpose. The reader can consult [19] for the other form. It should be noted that in [19], n is the time index, and k the constraint index, whereas we use k for the time index.

The set of linear constraints on the weight vector can be concisely written in the form

$$\mathbf{C}^T \mathbf{w} = \mathbf{f}$$

where **C** is the matrix whose columns are the coefficients of the weights in each constraint.

The optimization problem is

$$\text{Minimize} \qquad J = \frac{1}{4} < (|y|^2 - d)^2 >$$

$$\text{subject to} \qquad \mathbf{C}^T\mathbf{w} = \mathbf{f}$$

$$\text{where} \qquad y = \mathbf{x}^T\mathbf{w}$$

and we have replaced δ^2 by d

To remove the constraint, first note that

$$\mathbf{w}_q = \mathbf{C}^* (\mathbf{C}^T\mathbf{C}^*)^{-1} \mathbf{f}$$

is a weight vector that satisfies the constraints, i.e.,

$$\mathbf{C}^T\mathbf{w}_q = \mathbf{f}$$

Any other weight vector **w** which satisfies the constraints can be written in the form

$$\mathbf{w} = \mathbf{w}_q + \mathbf{w}_n$$

where $\mathbf{w}_q$ is in the null space of $\mathbf{C}^T$, i.e., $\mathbf{C}^T\mathbf{w}_q = 0$. The optimization for the LC-CMA takes place over $\mathbf{w}_n$. Although $\mathbf{w}_n$ is a vector of length N, it belongs to a subspace of dimension less than N (actually N - rank(**C**)). This information can be incorporated explicitly into the algorithm. Let $\mathbf{W}_s$ be a matrix whose column vectors form a basis for the null space of $\mathbf{C}^T$. Write

$$\mathbf{w}_n = \mathbf{W}_s\mathbf{w}_a$$

then

$$\mathbf{w} = \mathbf{w}_q + \mathbf{W}_s\mathbf{w}_a$$

The LC-CMA becomes

$$\text{Minimise} \qquad J = \frac{1}{4} < (|y|^2 - d)^2> \qquad \text{over } \mathbf{w}_a$$

$$\text{where} \qquad y = \mathbf{x}^T\mathbf{w}_q + \mathbf{x}^T\mathbf{W}_s\mathbf{w}$$

Applying the steepest descent method to this yields update equations

$$\mathbf{x}_a = \mathbf{x}^T\mathbf{w}_s$$
$$\mathbf{w}_a = \mathbf{w}_a(k) - \mu\, (|y(k)|^2 - d)\; y(k)\; \mathbf{x}_a^*(k)$$

A block diagram of this generalized sidelobe canceler is given in Figure 9.

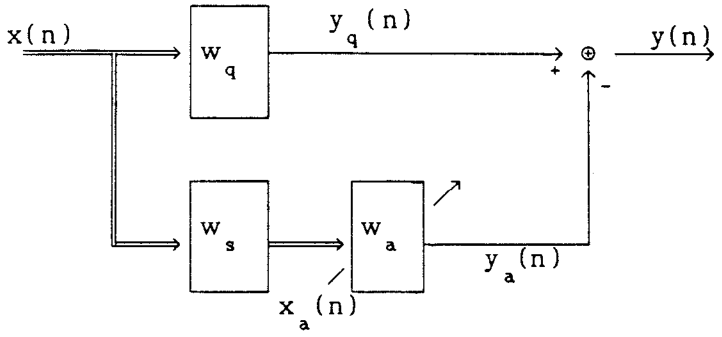


Figure 9. The Generalized Sidelobe Canceller Structure (from [19])

Rude and Griffiths in [19] give simulation results for a four element square array, receiving a QPSK signal from a known direction, together with a half baud delay multipath component, two CM interferers and noise. They compare the CMA, LC-CMA and linearly constrained minimum power algorithm. About 8000 samples are needed for convergence of each algorithm. They report that the LC-CMA locked on to the SOI consistently, whilst the CMA suffered from capture, and the LCMP often phased the multipath components together, causing cancellation of the SOI.

The LC-CMA cost function contains the parameter d, which represents the square of the modulus of the SOI. In the CMA, this was set to 1, in effect defining the filter gain. However, if constraints are present, it may no longer be possible to scale the weights, without violating the constraints. Hence the LC-CMA simulation as presented also includes an assumption that the power of the SOI is known. In the case of unknown SOI power, Ferrara suggested tuning the gain, until the SOI is found. Rude and Griffiths suggest that it might be possible to make the factor d adaptive, and carry out the minimization over $\mathbf{w}_a$ and d, using a steepest descent algorithm for d:

$$d(k+1) = d(k) + \mu\,(|y(k)|^2 - d(k))$$

They report having successfully simulated this in the same environment as the previous simulation, with convergence to the desired signal, independent of the starting point for d, taking about 16,000 samples.

4.5 Hybrid Schemes

In an effort to improve convergence rates, or avoid capture effects, some researchers have proposed combining the CMA with another algorithm, with mixed results.

Satorius, et al, in [20], examined a hybrid of the CMA and the adaptive linear prediction (ALP) filter. The performance of the hybrid was erratic, and, in many cases, worse than that of the CMA alone. They also compared the CMA with the ALP, and concluding that the CMA provided a definite performance advantage over the ALP, and suggested its suitability for the rejection of interference in direct sequence spread spectrum communications systems.

On the other hand, Goldberg and Iltis modelled a hybrid of the CMA and the recursive least squares-decision directed algorithm [21]. In their simulations, this hybrid performed better than the CMA, and was able to handle situations where the CMA was totally ineffective.

5 ALTERNATIVE OPTIMIZATION METHODS

The extensions to the standard CMA mentioned in the previous section all use the method of steepest descent to find a minimum of the cost function. In this section we will report on three alternative optimization methods that have appeared in the literature. Two of these are very similar, and indeed, appear almost identical when applied to an array. What is surprising is that they were developed from different starting points.

5.1 Least Squares CMA

Agee in [22] and [18] has presented a variant of the CMA which uses the method on non-linear least squares (Gauss' Method) to optimize the CMA cost function. His algorithm is developed for an array of antenna elements, with two port polarization combiners in mind. Consequently there are three indices involved: time index (k), update index (i), and array element index (n). The time index and update index are not identified. The algorithm can then be implemented is a block update form, in which the statistics required to calculate the weight update are computed over each data block, and used to set the weights for the next data block. Block update may represent a trade off between an algorithm which converges faster than the standard CMA, but which requires more calculations and therefore more processor time.

Presentation of Agee's algorithm is complicated by the fact that his two papers only sketch the details of the algorithm, and his notation is not consistent between them. In particular, n in [18] is the antenna index, but in [22] the role of n is not clearly defined. If [22] is addressing the same context as [18], n there would be the time index (in [22] equation 6, together with the block update form of **w** suggest this). This is the version we shall present, but we shall change the notation to remain consistent with that for the original CMA, i.e., the time index is k, update i and element n. (Agee uses k for the weight update in [22] and [18].)

For any particular output signal y,

$$y(k,i) = \mathbf{w}^T(i)\mathbf{x}(k)$$

where $\mathbf{x}(k)$ is the input data vector for the array at time k, and $\mathbf{w}(i)$ is the weight vector at stage i. The weights **w** may depend on the output channel. The method of non-linear least squares involves minimizing a sum of squares of non-linear functions. To arrange a CMA type cost function in this form, it is necessary to choose $q=2$, and estimate the expectation by a sum. For $p=1$ the cost function is

$$< (|y(k,i)| - 1)^2 > \; = \; < (|\mathbf{x}^T(k)\mathbf{w}(i)| - 1)^2 >$$

where the expectation is taken over time, k. Estimating the expectation as the average over the last data block, and dropping the constant in the denominator gives as cost function

$$\sum_k (|\mathbf{x}^T(k) \; \mathbf{w}(i)| - 1)^2$$

The update equation for the method of non-linear least squares is

$$\mathbf{w}(i{+}1) = \mathbf{w}(i) - (\mathbf{X}^*\mathbf{X}^T)^{-1} \mathbf{X}^* \; (\mathbf{y}(i) - \mathbf{d}(i))$$

where **X** is the input data matrix whose kth column is the data vector at time k, **x**(k), **y**(i) is the vector whose kth element is the result of applying **w**(i) to **x**(k) and **d**(i) is the complex limited output of y(i), whose elements are y(i,k)/|y(i,k)| . The update equation can be simplified to

$$\mathbf{w}(i{+}1) = (\mathbf{X}^*\mathbf{X}^T)^{-1} \mathbf{X}^* \; \mathbf{d}(i)$$

An alternative form, given in [18] is

$$\begin{aligned} \mathbf{w}(i{+}1) &= < \mathbf{x}^*(k) \; \mathbf{x}^T(k) >^{-1} \; < \mathbf{x}^*(k) \; \mathbf{d}(i) > \\ &= \mathbf{R}^{-1} \; \mathbf{g}(i) \end{aligned}$$

where **R** is the data autocorrelation matrix and **g** is the gain aperture vector computed using **w**(i).

In [22] Agee states inequalities which imply global convergence of the LS-CMA to a cost function stationary point. In [18] he states that the algorithm is gain invariant, in the sense that if **w**(i) updates to **w**(i+1), then any amplitude modulated translation of **w**(i) will also update to **w**(i+1), i.e., is α(k) is any non-negative real function, the the weights **w**(i) and α(k)**w**(i) both update to the same weights **w**(i+1). This property is a consequence of the update equation for **w**, where the old weights only occur in **d**, which is limited to have modulus one, thereby cancelling out any amplitude changes to **w**. Gain invariance may have important consequences in an implementation of the LS-CMA, as it may be possible to scale the weights to remove complex divisions in the weight update equations, and so lower the complexity of the calculations.

Agee in [18] also mentioned that the LS-CMA can be extended to a real version, which adapts a set of real weights to convert a real IF signal vector and to a multitarget LS-CMA, which processes multiple SOIs on a parallel basis. According to Agee, the real weight update equation is the same as the complex weight update equation, without the conjugation and with **d** replaced by the signum function. The multitarget LS-CMA uses the complex weight update equation, with added orthogonalisation constraints of the form

$$< \mathbf{y}_n(k,i{+}1) \; \mathbf{y}_m^*(k,i{+}1) > \; = 0$$

Block diagrams and architectures for a memoryless two part polarization combiner are also given in [18].

Agee has simulated the LS-CMA for a two sensor linear array in an environment which included a co-channel interferer, and noise at various levels. He has compared the performance of two forms of the LS-CMA - a static form, where the same data block is repeatedly used, and a dynamic form, with changing data blocks - and the SD-CMA. A 32 sample data block was used. The dynamic LS-CMA outperformed the standard CMA in his simulations. It was resistant to capture and regularly converged at a faster rate, varying from 2-3 times in [18] to 10 times in [22] and up (e.g., where the SD-CMA was captured by noise in a severe interference environment, and had not converged in 25000 samples, the dynamic LS-CMA converged in 320 samples).

The LS-CMA was formulated using p = 1. In [11] there is a brief reference to some work of Smith on applying the method of non-linear least squares when p = 2. It is reported that the preliminary simulations suggest poor convergence properties.

5.2 Orthogonalised or Lattice CMA

It was mentioned before that the CMA algorithm has many similarities with the Least Mean Squares Algorithm. Gooch and Lundell have used the analogy between the two algorithms to suggest a version of the CMA algorithm with improved convergence properties [17]. Noting that the Recursive Least Squares technique can be successfully applied to the LMS algorithm, they proposed that an analogous change be made to the CMA algorithm for p = q = 2. Specifically, they suggested pre-multiplying the data vector by the inverse of the correlation matrix. This is equivalent to pre-orthogonalising the input correlation function [11]. For this reason, they called their algorithm the Orthogonalized CMA. It is remarkable that the O-CMA turns out to be virtually identical to the LS-CMA in the case of an array.

Gooch and Lundell have chosen to represent the action of the filter in terms of the hermitian transpose. Since their formulation involves several formulae, we shall not alter this in our exposition, lest errors slip in. However, we shall continue to use k for the time index, where they use n. As in the LS-CMA, **x**(k) is the vector of data samples from the array elements at time k. For the standard CMA, the processor output and weight update equations are

$$y(k) = \mathbf{w}^H(k)\ \mathbf{x}(k)$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\ \mathbf{x}(k)\ \varepsilon^*(k)$$

$$\varepsilon(k) = y(k) - y(k)|y(k)|^2$$

For the O-CMA, the weight update equation is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \mathbf{R}^{-1}(k+1)\ \mathbf{x}(k)\ \varepsilon^*(k)$$

where $\mathbf{R}^{-1}(k+1)$ is an estimate of the correlation of the input data. This matrix can be updated recursively using the formula

$$\mathbf{R}^{-1}(k+1) = \frac{1}{1-\alpha}\,\mathbf{R}^{-1}(k) - \frac{1}{1-\alpha}\left[\frac{\alpha\mathbf{R}^{-1}(k)\mathbf{x}(k)\mathbf{x}^{H}(k)\mathbf{R}^{-1}(k)}{(1-\alpha) + \alpha\mathbf{x}^{H}(k)\mathbf{R}^{-1}(k)\mathbf{x}(k)}\right]$$

In [17] Gooch and Lundell reported on simulations of the OCMA for a four element array in a multipath environment. They found that it converged more than 8 times faster then the SD-CMA. They also performed a capture analysis on the algorithm for a two tone situation, similar to that analyzed in [9]. Their analysis showed that the O-CMA would capture the signal which was initially strongest at the array output, and null the other. Thus the O-CMA is also more predictable than the SD-CMA in its capture performance.

The formulae for the O-CMA weight update does not rely on the data necessarily coming from an array. The vector $\mathbf{x}(k)$ could contain the time delayed samples from a single antenna. This situation was addressed in [11] by Gooch, Ready and Svoboda. The core of that paper lay in the formulation of an adaptive lattice filter that provides an efficient method for orthogonalising the input signal in only order(N) operations per sample. The lattice structure is illustrated in Figure 10. Associated with each step in the filter is a reflection coefficient $\kappa_m(k)$, and forward and backward prediction residuals $f_m(k)$ and $b_m(k)$, where the index m runs over the stages of the filter (m = 0 being the most recent), and, as before, k is the time index. Gooch, Ready and Svoboda used a gradient search method for estimating the reflection coefficient.

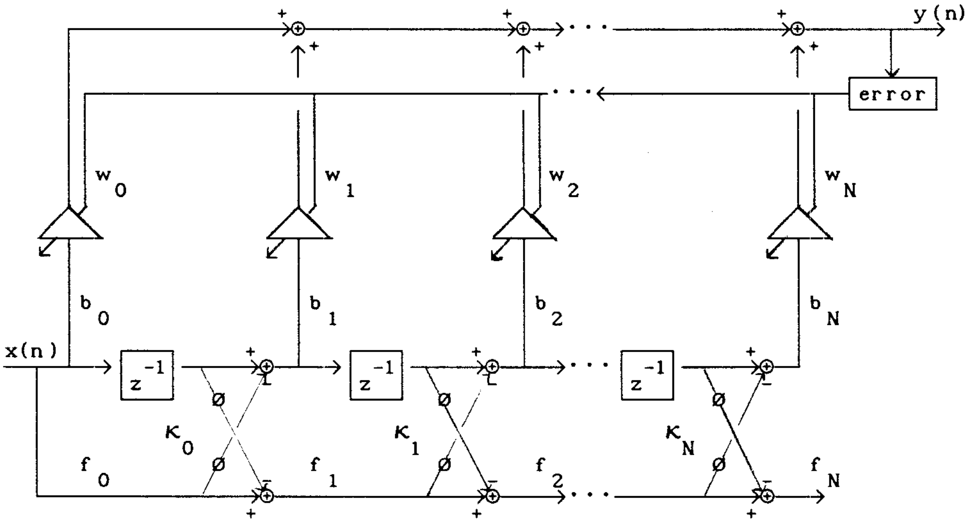


Figure 10. Block diagram of Lattice CMA (from [11])

The update equations given in [11] for the lattice parameters are

$$f_{m+1}(k) = f_m(k) - \kappa_m(k)\, b_m(k-1); \qquad f_0(k) = x(k)$$

$$b_{m+1}(k) = b_m(k-1) - \kappa^*_m(k)\, f_m(k); \qquad b_0(k) = x(k)$$

$$\kappa_m(k+1) = \kappa_m(k) - \frac{\mu}{d^2_m(k)} \left[f_{m+1}(k) b^*_m(k-1) + fm(k) b^*_{m+1}(k) \right]$$

where

$$d^2_m(k) = \alpha d^2_m(k-1) + (1-\alpha)\left[|f_m(k)|^2 + |b_m(k-1)|^2 \right]$$

The update equation for the mth weight is

$$w_m(k+1) = w_m(k) + \frac{\mu_c}{g^2_m(k)}\, b_m(k)\, \varepsilon^*(k)$$

$$g^2_m(k) = (1-\alpha) g^2_m(k-1) + \alpha |b_m(k)|^2$$

Here ε is the usual CMA error, $d^2_m(k)$ is an estimate of the mth stage error signal power at time k, and $g^2_m(k)$ is an estimate of the power of $b_m(k)$. The step sizes $\mu/d^2_m(k)$ and $\mu_c/g^2_m(k)$ are chosen to depend inversely on the power, in order to maintain the same adaptive time constant for each reflection coefficient and filter weight respectively, and to ensure that adaptation time is independent of eigenvalue disparity.

Gooch, Ready and Svoboda presented examples of simulation results comparing the lattice CMA with the standard CMA, in an environment where a constant modulus signal was distorted by an interferer and additive noise. The LCMA appears to converge at least an order of magnitude faster than the SD-CMA in their tests. The filter length was not specified. One of their results is reproduced in Figure 11.

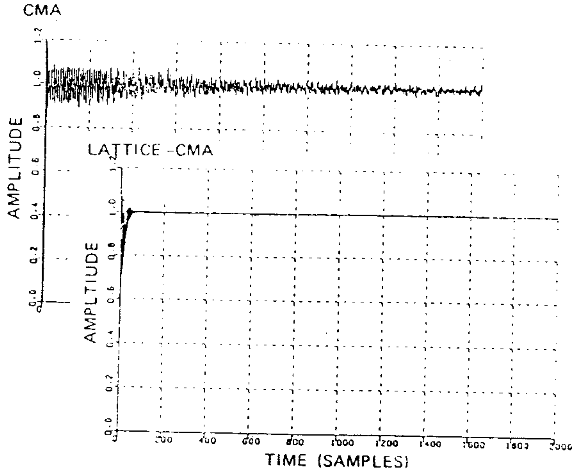


Figure 11. (a) Output modulus of lattice CMA and conventional CMA (from [11]).

5.3 Modified CMA

The CMA and its variants which have been discussed so far all perform their optimization over the weight of the filter. A modified form of the CMA has been introduced by Kammeyer, Mann and Tobergte [5, 23]. Their approach is to model the transmission channel using few parameters and then to optimize over the parameters. They claim that the number of variables is reduced from hundreds of weight coefficients to two parameters, and an overall filter length of about 30. The resulting algorithm has a significantly faster convergence rate than the SD-CMA. Their algorithm is developed for a multipath transmission channel only. They do not consider the case of independent interferers.

Their algorithm is based on a two-ray model of a multipath transmission channel,

$$x(k) = s(k) + r(k)s(k-\kappa)$$

where x(k) is the received signal at time k, s is the transmitted signal r is the reflection coefficient and κ the relative time delay. Note that r and κ are time varying functions. The authors show that any multipath channel can be formulated in this way, provided the parameters are allowed to depend on time. For simplicity, however, in the rest of their exposition, they assume that r and κ are independent of time. They present experimental results for the time varying case.

The transfer function for a two-ray multipath transmission channel can be written as

$$C_0(z) = 1 + r\ z^{-\kappa}$$

The inverse is, of course, unstable for $\kappa < 0$, but can be approximated by a series

$$H(z) = \sum_{\nu=0}^{n-1} (-r)^{\nu}\ z^{-\nu\kappa}$$

provided $|r| < 1$. The filter is non-causal if $\kappa < 0$, but can be made causal if a time shift of

$$n_0 = (n-1)\ \max\{\ |\kappa(t)|\ \}$$

sampling intervals is introduced.

Implementation of this inverse filter could be unwieldy in practice if n is large, or if κ is not an integer. Kammeyer, Mann and Tobergte introduce two modifications to reduce hardware complexity. The first is an interpolation scheme to approximate fractional values of κ. The second is a cascade of inverse filters of the type above. The combination of channel and approximate inverse filters has transfer function

$$1 \;-\; (-r)^n \, z^{-n\kappa}$$

This can be regarded as a new multipath channel with increased delay and reduced reflection factor. A series of approximate inverse filters can be cascaded until the reflection factor is sufficiently small. In this cascade, the value of n for each filter can be fixed. Kammeyer, Man and Tobergte use n = 2, and typically consider N = 6 filters in their simulations.

Since the filter weights are simple functions of r and κ (in the time varying case), the optimization of the CMA cost function need take place over these two parameters only. Kammeyer, Mann and Tobergte use the p = q = 2 version of the CMA cost function and derive the update equations for r and κ.

Simulations results given in [5] suggest that the M-CMA is more effective at correcting multipath distortion than the other versions of the CMA. For example, in a static two-ray simulation, the M-CMA, with an overall FIR filter length of 32, converged in approximately 500 samples. A sample of the results in [5] is reproduced in Figure 12. The algorithm has also been realized in hardware. Results from a test with a stereo signal are shown in Figure 13.

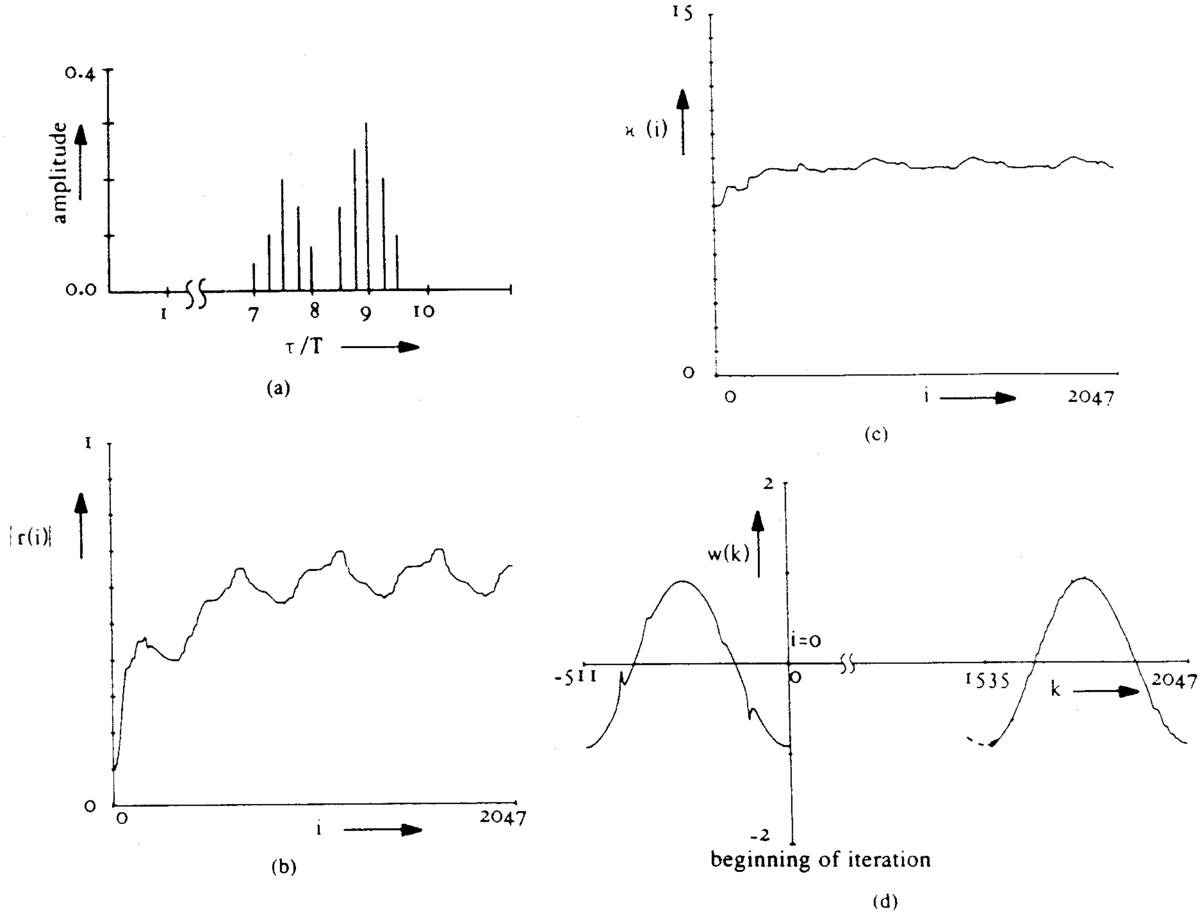
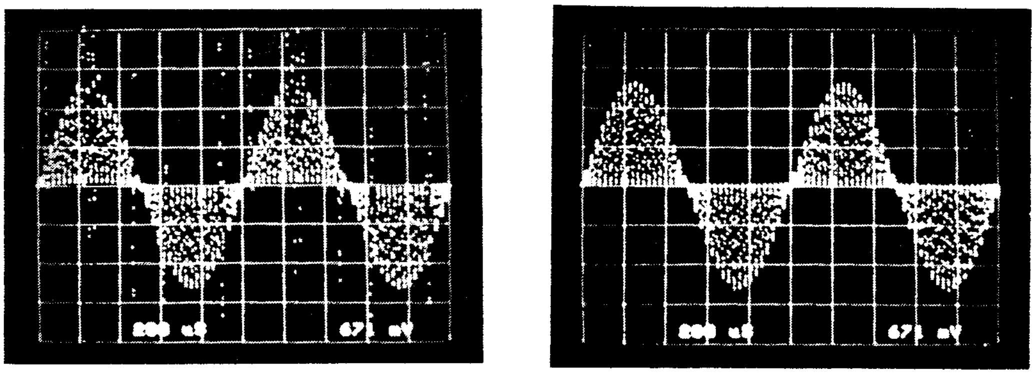


Figure 12. An example of M-CMA for multipath conditions (10 ray path). (a) Echo configuration. (b) Tracking performance of the refelction coefficient. (c) Tracking performance of the delay time. (d) Demodulated signal. (from [5])



(a) (b)

Figure 13. Received stereo signal under multipath (left channel: 1KHz tone). (a) Without equalizer. (b) Cascade equalizer with 6 complex coefficients. (from [5])

6 CONCLUSION

The concept of using the constant modulus of the transmitted signal as the cost function in an optimization method appears to be an effective way of creating an adaptive filter. The usefulness of the different optimization methods varies. It would seem that the most promise for fast convergence rates lies with pre-orthogonalising the data, or modeling the channel parameters (i.e. with the O-CMA or M-CMA). Furthermore, it is likely that the extensions of the SD-CMA presented in Section 4 might be also applied to these two methods. It is not clear that the M-CMA would be effective in dealing with co-channel interferers. Finally any optimization routine in general is likely to suffer from capture effects, due to the non-convexity of the cost function. Use of additional information or "tuning" of parameters, whether by hand or adaptively, may be the only way to overcome this.

REFERENCES

[1] J. R. Treichler and B. G. Agee, "A new approach to multipath correction of constant modulus signals," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-31, pp. 459-472, Apr. 1983.

[2] D. N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. Commun.*, vol. COM-28, pp. 1867-1875, Nov. 1980.

[3] M. G. Larimore and J. R. Treichler, "Convergence behaviour of the constant modulus algorithm," in *Proc ICASSP 83, Boston*, vol. 1, 1983, pp. 13-16.

[4] J.O. Smith and B. Friedlander, "Global convergence of the constant modulus algorithm," in *Proc. ICASSP '85*, vol 3, 1985, pp. 1161-1164.

[5] K. D. Kammeyer, R. Mann and W. Tobergte, "A modified adaptive FIR equalizer for multipath echo cancellation in FM transmission," *IEEE Journ. on Sel. Areas in Comm.*, vol SAC-5, pp. 226-237, Feb., 1987.

[6] J. R. Treichler and M. G. Larimore, "Convergence rates for the constant modulus algorithm with sinusoidal inputs," in *Proc ICASSP 85*, vol. 3, 1985, pp. 1157-1160.

[7] M. G. Larimore and J. R. Treichler, "Noise capture properties of the constant modulus algorithm," in *Proc ICASSP 85*, vol. 3, 1985, pp. 1165-1168.

[8] B. G. Agee, "Convergent behaviour of modulus-restoring adaptive arrays in Gaussian interference environments," in *Proc. 22nd Asilomar Conf on Signals, Systems and Computers,* 1989, pp. 818-822.

[9] J. R. Treichler and M. G. Larimore, "The tone capture properties of CMA_based interference suppressors," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-33, pp. 946-958, Aug. 1985.

[10] J. Treichler and M. G. Larimore, "CMA-based techniques for adaptive interference rejection," in *Milcom 86*, vol. 3, 1986, pp. 47.3.1-5.

[11] R. Gooch, M. Ready and J. Svoboda, "A lattice-based constant modulus adaptive filter," in *Proc 20th Asilomar Conf. on Signals, Systems and Computers*, 1987, pp. 282-286.

[12] J. R. Treichler and M. G. Larimore, "A real-arithmetic implementation of the constant modulus algorithm," in *Proc ICASSP 84*, vol. 1, 1984, pp. 3.2.1-4.

[13] J. R. Treichler and M. G. Larimore, "New processing techniques based on the constant modulus adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-33, pp. 420-431, Apr. 1985

[14] C. R. Johnson, Jr., S. Dasgupta, and W. A. Sethares, "Averaging analysis of local stability of a real constant modulus algorithm adaptive filter," *IEEE Trans. Acoust., Speech, Signal Proc.,* vol. ASSP-36, pp.900-901, June 1988.

[15] E. R. Ferrara, "A method for cancelling interference from a constant envelope signal," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-33, pp. 316-319, Feb. 1985.

[16] W. A. Sethares, G. A. Rey and C. R. Johnson, "Approaches to blind equalization of signals with multiple modulus," in *Proc. ICASSP 89*, vol. 2, 1989, pp. 972-975.

[17] R. Gooch and J. Lundell, "The CM array: an adaptive beamformer for constant modulus signals," in *Proc. ICASSP 86, Tokyo*, vol. 4, 1986, pp. 2523-2526.

[18] B. G. Agee, "Fast adaptive polarization using the least-squares constant modulus algorithm," in *Proc. 20th Asilomar Conf. on Signals, Systems and Computers*, 1987, pp. 590-595.

[19] M. J. Rude and L.J. Griffiths, "Incorporation of linear constraints into the constant modulus algorithm," in *Proc. ICASSP 89*, vol. 2, 1989, pp. 968-971.

[20] E. Satorius, S. Krishnan, X-L. Yu, L. J. Griffiths, I. S. Reed, and T-K. Truong, "Suppression of narrowband interference via single channel adaptive preprocessing," in *Proc 22nd Asilomar Conf on Signals, Systems and Computers*, 1989, pp. 270-3.

[21] S. H. Goldberg and R. A. Iltis, "Joint channel equalization and interference rejection using a modified constant modulus algorithm," in *Milcom 88*, 1988 vol. 1, pp. 97-101.

[22] B. G. Agee, "The least-squares CMA: a new technique for rapid correction of constant modulus signals," in *Proc. ICASSP 86, Tokyo*, vol. 2, 1986, pp. 953-956.

[23] R. Mann and K. D. Kammeyer, "A pole-zero-tracking constant modulus algorithm," in *Proc. ICASSP 89*, vol 2, 1989, pp. 1227-1229.

Other Works on the CMA

R. P. Gooch and J. C. Harp, "Blind channel identification using the constant modulus adaptive algorithm," in *Proc. 1988 IEEE ICC*, vol 1., 1988, pp. 75-79.

R. A. Iltis and S. H. Goldberg, "Joint interference rejection/channel equalization in DS spread-spectrum receivers using the CMA equalizer and maximum-likelihood techniques," in *Milcom 87*, vol. 1, 1987, pp. 0109-0113.

N. K. Jablon, "Joint blind equalization, carrier recovery, and timing recovery for 64-QAM and 128-QAM signal constellations," in *Proc IEEE ICC 89*, vol. 2, 1989, pp. 1043-1049.

C. R. Johnson, Jr, "Admissibility in blind adaptive channel equalization," in *IEEE Control Systems Magazine,* vol. 11, no. 1, 1991, pp. 3-15.

M. G. Larimore and M. J. Goodman, "Implementation of the constant modulus algorithm at RF bandwidths," in *Proc. 19th Asilomar Conf. on Circuits, Systems and Computers*, 1986, pp. 626-629.

M. G. Larimore and J. R. Treichler, "Data equalization based on the constant modulus adaptive filter," in *Proc ICASSP 86, Tokyo*, vol. 2, 1986, pp. 949-952.

J. R. Treichler, S. L. Wood and M. G. Larimore, "Convergence rate limitations in certain frequency-domain adaptive filters," in *Proc. ICASSP 89*, 1989, pp. 960-963.

Page Classification
UNCLASSIFIED

Privacy Marking/Caveat
N/A

Department of Defence

DOCUMENT CONTROL DATA SHEET

1a. AR Number
AR-006-756

1b. Establishment Number
ERL-0568-RR

2. Document Date
AUG 91

3. Task Number
90/085

4. Title

CONSTANT MODULUS ALGORITHM
TECHNIQUES FOR THE
ENHANCEMENT OF SIGNAL QUALITY

5. Security Classification

U Document U Title U Abstract

S (Secret) C (Confi) R (Rest) U (Unclass)

* For UNCLASSIFIED docs with a secondary distribution LIMITATION, use (L)

6. No. of Pages 30

7. No. of Refs. 23

8. Author(s)

Peter L. Trudinger

9. Downgrading/Delimiting Instructions

N/A

10a. Corporate Author and Address

Electronics Research Laboratory
PO Box 1600
SALISBURY SA 5108

10b. Task Sponsor

DEFENCE

11. Officer/Position responsible for

Security..........................N/A..

Downgrading..................N/A..

Approval for Release.....DIRECTOR/ERL...................

12. Secondary Release of this Document

APPROVED FOR PUBLIC RELEASE

Any enquiries outside stated limitations should be referred through DSTIC, Defence Information Services, Department of Defence, Anzac Park West, Canberra, ACT 2600.

13a. Deliberate Announcement

NO LIMITATION

13b. Casual Announcement (for citation in other documents)

☑ No Limitation

□ Ref. by Author & Doc No only

14. DEFTEST Descriptors

ALGORITHMS; SIGNALS; FILTERS; CONSTANT MODULUS ALGORITHM; SIGNAL QUALITY

15. DISCAT Subject Codes

2014

16. Abstract

THE CONSTANT MODULUS ALGORITHM IS AN ADAPTIVE ALGORITHM FOR SELECTING EQUALISER FILTER WEIGHTS WHICH ATTEMPTS TO CORRECT FOR DEGRADATION IN SIGNAL QUALITY SUFFERED BY CONSTANT ENVELOPE SIGNALS (eg. FM) OVER AN IMPERFECT TRANSMISSION CHANNEL. THIS REPORT REVIEWS AND EVALUATES THE STANDARD CMA AND SEVERAL OF ITS VARIANTS, WHICH WERE DESIGNED TO IMPROVE ON ITS PERFORMANCE OR EXTEND ITS APPLICABILITY. TWO VARIANTS SEEM TO SIGNIFICANTLY INCREASE CONVERGENCE RATE AND APPEAR WORTHY OF FURTHER STUDY OR IMPLEMENTATION.

Page Classification UNCLASSIFIED
Privacy Marking/Caveat N/A

16. Abstract (CONT.)

17. Imprint

Electronics Research Laboratory
PO Box 1600
SALISBURY SA 5108

18. Document Series and Number	19. Cost Code	20. Type of Report and Period Covered
ERL-0568-RR	791671	RESEARCH REPORT

21. Computer Programs Used

N/A

22. Establishment File Reference(s)

N/A

23. Additional information (if required)

N/A